Remarks

In the Office Action dated August 24, 2004, the Examiner rejected claims 21, 22, 27, 29, and 30 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,477,562 to Nemirovsky ("Nemirovsky) and rejected claims 1-20, 23-26, and 28 under 35 U.S.C. § 103(a) under 35 U.S.C. § 103(a) in view of Nemirovsky and an excerpt from Computer Organization and Design, The Hardware/Software Interface, by John Hennessy et al. ("Hennessy"). Further, in the Office Action, the Examiner objected to the drawings as including reference numerals not mentioned in the description; objected to claims 1 and 16 on informalities; objected to claim 2 as being of improper dependent form; and rejected claim 25 under 35 U.S.C. § 112, second paragraph, as being indefinite.

By this Amendment, Applicants have amended the specification to correct a typographical error and amended claims 1, 2, 6, 10, 14, 16, 19, 21, 25, 27, and 28 for form. Claim 5 has been cancelled without prejudice or disclaimer. New claim 31 has been added.

Regarding the objection to the drawings, Applicants have amended the specification to reference element 607. Accordingly, the objection to the drawings has been obviated.

The Examiner also objected to claims 1, 2, and 16. Although Applicants do not necessarily agree with all of the Examiner's objections, in the interest of expediting prosecution, Applicants have amended these claims. In view of the

these claim amendments, Applicants submit that the objections have been obviated.

The Examiner rejected claim 25 under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite. Applicants submit that claim 25, as amended, is clear and definite and fully complies with 35 U.S.C. § 112.

CLAIM REJECTIONS UNDER 35 U.S.C. § 102(e) BASED ON NEMIROVSKY

Claims 21, 22, 27, 29, and 30 stand rejected under 35 U.S.C. § 102(e) based on Nemirovsky. Applicants respectfully traverse this rejection.

A proper rejection under 35 U.S.C. § 102 requires that a reference teach every aspect of the claimed invention either explicitly or impliedly. Any feature not directly taught must be inherently present. See M.P.E.P. § 2131. Nemirovsky does not disclose or suggest the combination of features recited in Applicants' claims 21, 22, 27, 29, and 30.

Claim 21, as amended, is directed to a method for processing a packet to determine control information for the packet. The method includes reading a plurality of instructions, generating a predicted address based on a predetermined one of the read instructions, evaluating the read instructions, and selecting one of the read instructions based on the evaluations. Further, the method includes performing operations related to determining the control information for the packet based on the selected instruction, the operations including generating a next address for reading instructions.

Nemirovsky is directed to prioritized instruction scheduling for multi-streaming processors. The multi-streaming processor of Nemirovsky has multiple streams for processing multiple threads. (Nemirovsky, Abstract). According to Nemirovsky, the various streams are assigned priority codes, which are used to determine relative access of the streams to resources as well as which stream has access at any point in time. (Nemirovsky, Abstract).

Although Nemirovsky discloses the execution of multiple instruction streams, the multiple instruction streams appear to generally be independent of one another and are executed on a priority basis. This aspect of Nemirovsky is described in detail at column 7, lines 12-27. Nemirovsky states:

> The net effect of the queues is that there are concurrent streams of instructions from which eligible instructions may be issued to functional resources. Each stream that the processor is equipped to execute has a context frame containing the program counter and register file for that stream. A thread is made active by loading an available context frame with the thread's program counter address and register values and by assigning it an active priority. There may be only a single thread to be executed, in which case there is a single stream of instructions to execute. When there are more active threads than streams available to execute threads, a number of threads up to the available number of context frames are made active and the remaining threads remain temporarily inactive. It is typically a function of an operating system to assign threads to streams of a multi-streaming processor.

(Nemirovsky, col. 7, lines 12-27). As described in this section, each instruction stream has a program counter and a register file. When a thread is to be made active, its "context" is loaded and the thread begins to execute. Once a thread is selected and made active, the instructions in the thread appear to execute in a conventional manner.

Claim 21, in contrast to Nemirovsky, discloses, for example, reading a plurality of instructions, evaluating the read instructions, and selecting one of the read instructions based on the evaluations. As is further recited in claim 21, operations are performed based on the read instruction. Nemirovsky completely fails to disclose or suggest evaluating read instructions and selecting one of the instructions based on the evaluations, as recited in claim 21.

In rejecting claim 21, the Examiner points to column 5, lines 10-21 of Nemirovsky as allegedly disclosing the evaluation of read instructions, stating that this section shows "that the priority (evaluation results) is dynamically determined or evaluated and is based on each stream and thus the instructions therein." (Office Action, page 5). Most of this section of Nemirovsky is quoted above. Even assuming that Nemirovsky discloses evaluating the priority of streams, Nemirovsky in no way discloses or suggests evaluating a plurality of read instructions and selecting one of the read instructions based on the evaluations, as recited in claim 21.

In columns 8 and 9, Nemirovsky discusses various techniques for evaluating the priority of instruction streams. Nemirovsky discloses, for example, a static priority stream in which the streams appear to be pre-assigned "static" priorities. (Nemirovsky, col. 8, lines 8-18). The assigned static priorities are not based on the instructions within the streams, and are clearly not based on the evaluations of the instructions within a stream. At column 8, lines 19-51, Nemirovsky discloses a round-robin priority scheme, and at column 8, line 52 through column 9, line 41, Nemirovsky discusses a number of other possible

schemes for assigning priorities to instruction streams. None of these techniques, however, involve "evaluating the read instructions" and "selecting one of the read instructions based on the evaluations," as recited in claim 21. In contrast, the focus of Nemirovsky appears to be on using priorities to efficiently select a stream so that instructions within the stream can be evaluated. Thus, Nemirovsky selects a stream and then evaluates instructions within the selected stream. According to claim 21, however, instructions are first evaluated and then selected based on the evaluations.

Claim 21 further recites "generating a predicted address based on a predetermined one of the read instructions." The Examiner points to column 6, lines 9-16 of Nemirovsky as allegedly disclosing this feature of claim 21. Applicants respectfully disagree with the Examiner. This section of Nemirovsky states:

> Criteria for both access and priority determination may be from varied sources as well; in some cases according to on-chip statistics, such as current cache and memory requests, functional unit utilization or branch prediction, among other things; in others according to data arrival and availability, in others by input from off-chip, and in combinations of these and other criteria.

(Nemirovsky, column 6, lines 9-16). At most, this section of Nemirovsky states that criteria for access and priority determination may be based on a number of statistics, including statistics relating to branch prediction, but does not disclose any specific branch prediction technique. This section of Nemirovsky does not disclose generating a predicted address based on predetermined read instructions, as recited in claim 21.

For at least these reasons, Applicants submit that the rejection of claim 21 is improper and should be withdrawn. At least by virtue of its dependency on claim 21, the rejection of claim 22 under 35 U.S.C. § 102(e) is also improper and should be withdrawn.

Independent claim 27 and its dependent claims 29 and 30 were also rejected by the Examiner under 35 U.S.C. § 102(e) based on Nemirovsky. For the following reasons, Applicants respectfully traverse.

Claim 27, as amended, is directed to a processing device including means for simultaneously reading a plurality of processing instructions from instruction memory that relate to processing a packet and means for selecting one of the read instructions for execution based on a priority encoding of evaluation results related to each of the read instructions.

Nemirovsky does not disclose or suggest the features recited in claim 27. Nemirovsky, for instance, does not disclose "means for simultaneously reading a plurality of processing instructions from instruction memory that relate to processing of a packet." Although Nemirovsky discloses a multi-stream processor executing multiple threads, nothing in Nemirovsky discloses or suggests means for simultaneously reading a plurality of packet processing instructions from instruction memory that relate to processing a packet. The various instruction streams of Nemirovsky appear to be independent streams, each associated with its own processing context. Nemirovsky, therefore, cannot be said to disclose simultaneously reading a plurality of processing instructions from instruction memory that relate to processing of a packet.

Claim 27 further recites means for selecting one of the read instructions for execution based on a priority encoding of evaluation results related to each of the read instructions. Applicants submit that Nemirovsky also fails to disclose or suggest this feature of claim 27. As discussed previously, Nemirovsky discusses a number of possible schemes for assigning priorities to instruction streams. None of these techniques, however, involve selecting one of a plurality of read instructions based on a priority encoding of evaluation results related to each of the read instructions. Nemirovsky selects a stream and then evaluates instructions within the selected stream. Nemirovsky does not disclose, however, selecting an instruction (or stream) based on evaluation results related to each of a plurality of instructions.

For at least these reasons, Applicants submit that the rejection of claim 27 is improper and should be withdrawn. At least by virtue of its dependency on claim 27, the rejections of claims 29 and 30 under 35 U.S.C. § 102(e) are also improper and should be withdrawn.

CLAIM REJECTIONS UNDER 35 U.S.C. § 103(a) BASED ON NEMIROVSKY AND HENNESSY

Pending claims 1-4, 6-20, 23-26, and 28 stand rejected under 35 U.S.C. § 103(a) based on Nemirovsky in view of Hennessy. More particularly, in rejecting independent claim 1, the Examiner contends that Nemirovsky discloses many of the features recited in claim 1 but concedes that Nemirovsky does not disclose a plurality of memories nor pipelining. The Examiner contends, however, that one

of ordinary skill in the art would have found it obvious to modify Nemirovsky in view of Hennessy to obtain the claimed invention. Applicants disagree and respectfully traverse this rejection.

Claim 1, as amended, is directed to a pipelined processor comprising a first pipeline stage and a second pipeline stage. The first pipeline stage includes a plurality of instruction memories and a program counter corresponding to a location in each of the instruction memories from which instructions are read. The second pipeline stage includes an evaluation component corresponding to each of the instruction memories, the evaluation component generating evaluation results based on each of the instructions read from the instruction memories, a priority encoder configured to select one of the instructions based on the evaluation results generated from the instructions read from the instruction memories, and an execution unit configured to receive the selected one of the instructions and to perform operations indicated by the selected instruction.

Regarding the first pipeline stage recited in claim 1, which includes a plurality of instruction memories, the Examiner concedes that this feature is not disclosed by Nemirovsky but contends that "fetching multiple threads and streams from [a single] instruction cache memory is taught . . . the inclusion of a plurality of instruction cache memories to perform the same function as a single instruction cache memory provides no new or unexpected result over the prior art. Therefore, one of ordinary skill in the art would have found it obvious to duplicate the instruction cache memory." (Office Action, page 9).

Applicants respectfully disagree with the Examiner's reasoning. In addition to reciting a plurality of instruction memories, the first pipeline stage of claim 1 includes "a program counter corresponding to a location in each of the instruction memories from which instructions are read." Nemirovsky, in contrast, explicitly discloses that each stream "has a context frame containing a program counter and register file for that stream." (Nemirovsky, column 7, lines 14-16). Accordingly, even assuming, for the sake of argument, that the disclosure of Nemirovsky was extended to include a plurality of instruction memories, there would still be no disclosure or suggestion of the program counter of claim 1, which is recited as corresponding to a location in each of the instruction memories from which instructions are read. Instead, one or ordinary skill in the art modifying Nemirovsky to include multiple instruction memories would likely modify Nemirovsky to independently read from the instruction memories based on the program counters corresponding to each of the streams. Nothing in Nemirovsky or Hennessy discloses or suggests a program counter corresponding to a location in a plurality of instruction memories, as required by claim 1.

Claim 1, as amended, further recites an evaluation component corresponding to each of the instruction memories and a priority encoder configured to select one of the instructions based on evaluation results generated from the instructions read from the instruction memories. As previously discussed, Nemirovsky discusses a number of possible schemes for assigning priorities to instruction streams. None of these techniques, however, involve

selecting one of a number of instructions based on evaluation results generated from the instructions. Hennessy also fails to disclose or suggest this feature of claim 1.

The second pipeline stage of claim 1 further recites an execution unit configured to receive the selected one of the instructions and to perform operations indicated by the selected instruction. Applicants submit that because neither Nemirovsky nor Hennessy discloses or suggests selecting one of the instructions, as recited in claim 1, these references could not possibly disclose or suggest the execution unit of claim 1, which performs operations indicated by the selected instruction.

For at least these reasons, Applicants submit that neither Nemirovsky nor Hennessy, either alone or in combination, discloses or suggests the combination of features recited in claim 1. Accordingly, the rejection of this claim is improper and should be withdrawn. The rejection of claims 2-4 and 6-9 under 35 U.S.C. § 103(a) should also be withdrawn, at least by virtue of the dependency of these claims from claim 1.

Independent claim 10 was also rejected based on Nemirovsky and Hennessy. Claim 10, as amended, is directed to a network device that comprises a physical interface and a processing unit. The processing unit includes a pipelined packet processing engine that includes: a first pipeline stage configured to read a plurality of packet processing instructions relating to processing of a first packet from instruction memories per processing cycle, and a second pipeline stage configured to select one of the instructions for execution.

The Examiner points to column 7, lines 3-7, of Nemirovsky as allegedly disclosing reading a plurality of packet processing instructions. This section of Nemirovsky states: "Typically, under the control of multi-threaded fetch unit 203, instructions from instruction cache 202 are transferred into multi-threaded fetch unit 203, where they are stored in prefetch buffers, decoded and placed in one or more queues." This section of Nemirovsky relates to reading instructions from an instruction cache to one or more queues. Nothing in Nemirovsky discloses or suggests, however, reading a plurality of packet processing instructions relating to processing of a first packet from instruction memories per processing cycle, as recited in amended claim 10. Although Nemirovsky may generally be said to read a plurality of instructions, a plurality of instructions are not disclosed as being read per processing cycle. Further, nothing in Nemirovsky discloses or suggests that multiple threads of Nemirovsky can operate on a particular packet. Accordingly, Nemirovsky cannot be said to disclose or suggest reading a plurality of packet processing instructions relating to processing of a first packet.

Claim 10 further recites a second pipeline stage configured to select one of the instructions for execution. Because Nemirovsky does not read a plurality of packet processing instructions, as recited in claim 10, Nemirovsky could then not possibly disclose or suggest selecting one of the instructions for execution.

For at least the reasons given above, Applicants submit that Nemirovsky does not disclose or suggest many of the features recited in claim 10. Hennessy fails to cure the deficiencies of Nemirovsky. Accordingly, the rejection of claim 10 under 35 U.S.C. § 103(a) is improper and should be withdrawn. The rejection

of dependent claims 11-20, at least by virtue of their dependency on claim 10, should also be withdrawn.

Claims 11-20 recite additional features patentable over the references of record. Claim 13, for example, further defines the device of claim 10, and recites that the second pipeline stage includes a priority encoder configured to select the one of the packet processing instructions based on evaluation results generated from the packet processing instructions read from the instruction memories, and an execution unit configured to receive the selected one of the packet processing instructions and to perform operations indicated by the selected packet processing instruction. As previously discussed, Nemirovsky does not select from among multiple instructions based on evaluation results generated from the instructions. Accordingly, for this reason also, the rejection of claim 13 is improper and should be withdrawn.

New claim 31 has been added that depends from claim 10. This claim recites that the non-selected ones of the packet processing instructions are not executed. Applicants submit that nothing in Nemirovsky or Hennessy discloses or suggests this features of claim 31.

Claims 23-26, which depend from claim 21, and claim 28, which depends from claim 27, were also rejected under 35 U.S.C. § 103(a) based on Nemirovsky and Hennessy. At least by virtue of their dependency from claim 21 or 28, Applicants submit that the rejections of these claims are improper and should be withdrawn.

In view of the foregoing amendments and remarks, Applicants respectfully request the Examiner's reconsideration of this application, and the timely allowance of the pending claims.

To the extent necessary, a petition for an extension of time under 37 CFR 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 50-1070 and please credit any excess fees to such deposit account.

Respectfully submitted,

HARRITY & SNYDER, L.L.P.

By: _____
Brian E. Ledell
Reg. No. 42,784

11240 Waples Mill Road
Suite 300
Fairfax, Virginia 22030
(571) 432-0800

Date: October 29, 2004